Exploiting a Real-time Non-geolocation Data to Classify a Road Type with Different Altitudes for Strengthening Accuracy in Navigation

Thitivatr PatanasakPinyo*
Mahidol University, Salaya, 73170, THAILAND

Abstract

Most location-based applications for navigation purposes use geolocation data, i.e., a pair of a latitude and a longitude, to determine a real-time location of a handheld device (e.g., smartphones or tablets) that runs the applications. This can be implemented basically by requesting a pair of a latitude and a longitude from the device's sensor that receives geolocation data from satellites. However, telling a device's location by GPS sensor is sometimes impractical, especially when the device is in a vehicle on a road that shares exactly the same geolocation with other roads. Particularly, this is a scenario that there is a ground-level road along with another elevated road (e.g., a turnpike) which is very common in cities like Bangkok, Singapore, or Hong Kong. The geolocation data yield no clue whether or not a vehicle is running on a ground-level road. Since a pair of a latitude and a longitude can no longer be used in such scenario, we proposed a methodology to identify the correct location of both a device and a vehicle without any involvement of geolocation data by using a Random Forest classifier and real-time traffic data that are able to be captured by a handheld device as training features to train a classification model. A completed experiment and results after testing the model were reported in this article.

Key Words: Location-based applications; navigation; geolocation; altitude; random forest.

1 Introduction

A location-based application has become a primary tool for road traffic navigation. Most users run the application on a handheld device such as a smartphone or a tablet. Furthermore, some automobile manufacturers provide a simple user interface to connect a smart device with a car's screen via USB or Bluetooth. This highly supports a navigation task since a map can be displayed on a wide screen on a car's console, which is way more comfortable for a driver. Navigating a route via location-based applications relies on a device's location sensor, which is a part that tells an exact location of the device by a pair of a latitude and a longitude. The sensor retrieves a signal from the satellites and computes both a latitude and a longitude. When the device has both values, it displays the device's location on the map. This whole process of how a location-based application works fine and invulnerable. However, there is a scenario that causes a device to misinterpret a location [7, 9]. In a crowded city like Bangkok, Singapore City, or Hong Kong, it is very common to see a ground-level road along with an elevated road above it. Since both roads are exactly located on the same location, every pair of a latitude and a longitude that belongs to the ground-level road also belongs to the elevated one as well. This confuses a device to distinguish what road a vehicle is on and leads to an incorrect navigation that can result in a major detour. Figures 1 and 2 show examples of roads that are under this condition. Figure 1 shows Borommaratchachonnani (Bor-Rom-Ma-Raj-Cha-Chon-Na-Ni) Elevated Road which is a 15-meter elevated from the ground while Figure 2 shows Borommaratchachonnani Frontage Road. Both roads are located in the city of Bangkok, Thailand. These east-west roads are 16-kilometer long that have an east end in the downtown of Bangkok and a west end at the west border of Bangkok (Figure 3). We would like to see how an application navigated when we drove on the frontage road. Our starting point was at the west end of the frontage road. Before we started, we set the destination on an application to be a shopping mall located at the north of Bangkok, particularly, Central Plaza Westgate. The location-based application that we used was Google Maps for Android. The device that we used was Samsung Galaxy Note 5 (2015). It turned out to be that the application understood that we drove on the elevated road, so it navigated us to take the closest exit to leave the elvated road then make a u-turn. However, the navigation was a major detour (Figure 4). We could have made a left turn at the intersection to head right to the north but the application did not think we can do this because it thought that we were not on the ground-level road. We did the same experiment on another day and found out that the application still navigated incorrectly. It still understood that we were on the elevated road (Figure 5). For this example, if we followed a navigation from the application, the detour would cost us significantly around 10 kilometer.

One question could be asked if there is a way to tell an altitude of a device. An altitude is a metric that informs a y-distance from the sea level. Clearly, knowing an altitude technically

*Faculty of Information and Communication Technology. Email: thitivatr.pat@mahidol.edu.

Figure 1: Borommaratchachonnani Elevated Road (image source: *www.dailynews.co.th*).

Figure 2: Borommaratchachonnani Frontage Road (image source: *Google Maps Street View*).

could do the trick. A GPS sensor in a smart device (plus a barometer sensor) is capable of retrieving an altitude from the satellites as well although the device itself does not have a built-in altimeter. However, an issue about an accuracy might exist in some conditions. In general, we can expect around ±23 meter for vertical error (altitude), which is around 1.5 times greater than horizontal error (latitude and longitude) [6, 5]. Since a vertical difference of most elevated and ground-level roads are usually less than 20 meter, an altitude calculated via this method is not dependable. For instance, the vertical difference between Borommaratchachonnani Elevated Road (Figure 1) and Borommaratchachonnani Frontage Road (Figure 2) is about 15 meter. It is also quite impractical for smart device manufacturers to equip a device with an actual altimeter since a demand to know an altitude for most users is uncommon.

With the problem being raised, we proposed a geolocation-free solution to determine whether a vehicle/device was on an elevated road or a ground-level road by using a set of real-time traffic data to train a classification model. By the term "real-time traffic data", we considered every possible metric that an average smart device was able to capture by its built-in sensors/features plus some aggregated data from further computation process. This set of data is detailed in Section 3.

We ran the preliminary experiment by choosing these Borommaratchachonnani Elevated Road and Borommaratchachonnani Frontage Road to collect data [11]. This data gathering process was conducted by driving on each road multiple trips and enabling an Android device to run our own application that invoked all built-in sensors to collect real-time data corresponding to the sensor. We set up a driving schedule in a way that it covered most days (days of week), time (time of day), and direction bounds (eastbound and westbound) to prevent a bias that might occur from imbalanced or inadequate feature values. The application automatically logged a new instance of data every two seconds. We trained two classification models using Random Forest and Bagging (with REPTree as a base classifier), respectively, as a classifier. The first model resulted in 99.7563% accuracy. The second model resulted in 93.4315% accuracy.

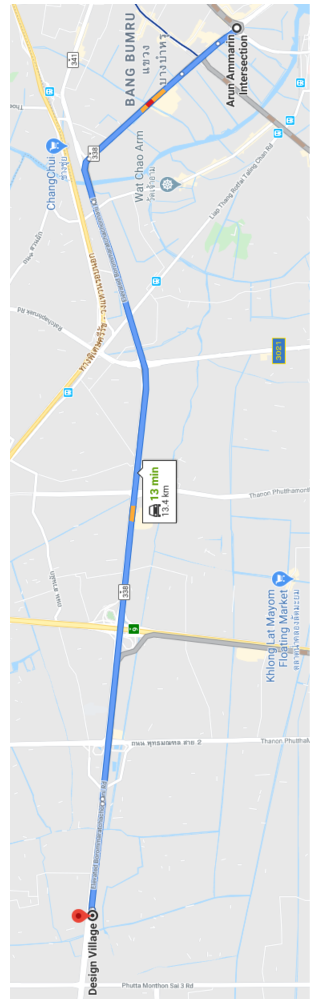


Figure 3: A layout of both the roads illustrated by Google Maps.

The structure of this paper is as follows. The first section introduces the problem scope with the proposed solution plus the results from the preliminary work. The second section refers to related work. Sections 3 and 4 discuss about a methodology and results obtained. The last section concludes this study and sheds some light to a possible future adaptation.

2 Related Work

It is quite true to say that a location-based application is necessary for most users of smart devices. A location-based application is capable of serving several purposes. Several

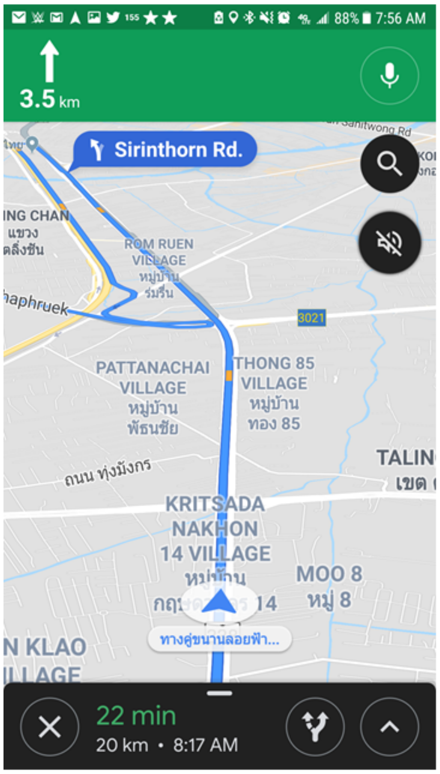


Figure 4: Navigated Routed from Google Maps on January 17, 2019.

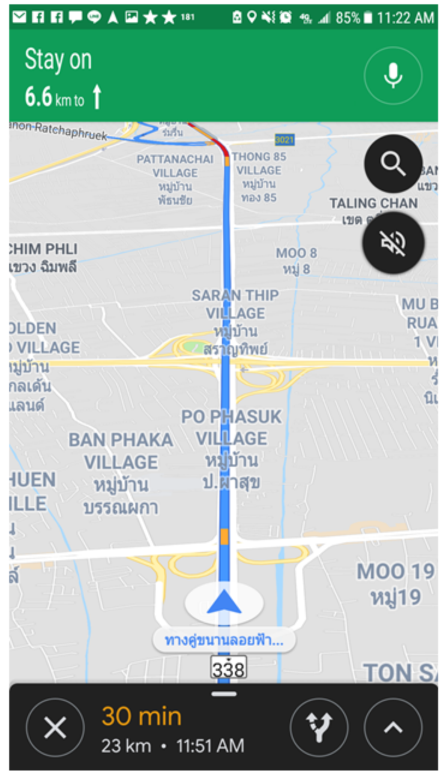


Figure 5: Navigated Routed from Google Maps on January 25, 2019.

studies integrated a location-based application with a census activity. Batinov et al. [1] proposed a detection pattern that can distinguish if a participant's spatial visualization (VZ) [8, 17] was low or high by analyzing their sequences of taps they made on the tablet screen while using it to perform address verification activity. Similarly, PatanasakPinyo et al. [10, 12, 14, 15, 16] did three studies that invited participants to verify addresses in the neighborhood using a location-based application on a tablet and found that there were some metrics that significantly could be used to identify a participant's spatial visualization such as a total number of pans, a total number of zooms, etc. PatanasakPnyo et al. [13] proposed a concept of empowering an indexing ability to a traditional raster map widely used by location-based applications. Sulaiman [19] verified that such metrics were still reliable even though an environment of an address verification task was changed from an actual neighborhood to a virtual reality. Whitney [20] enhanced an address verification task by combining a concept of a location-based application with a virtual reality. One popular activity that involved with location-based application is navigation. Most map applications such as Google Maps are designed and developed for the task of navigation with highly acceptable accuracy, which depended on a location sensor of a smart device (GPS) [3, 4, 18]. Lin et al. reported issues that were found when relying on GPS for navigation purposes [2]. Misinterpreting an exact location because of a difference in altitude is one difficulty users have to deal with when asking the application to navigate in cities with limited area [7, 9].

3 Methodology

We replicated the methodology previously implemented by PatanasakPinyo [11] by collecting all possible real-time traffic data that a device's sensor was able to retrieve while targeted driving on an actual road. The roads targeted for this study were Bangna - Chon Buri Expressway and Bangna – Trat Frontage Road. Both elevated and ground-level roads are east-west roads that located in Thailand (Figure 6). The methodology of this study consists of Study Design, Data Gathering, and Model Implementation.

3.1 Study Design

Since our objective was to develop a classification model that can correctly classify whether a vehicle is running on an elevated road or a ground-level road without involvement of any geolocation data (latitude, longitude, altitude), we have to include metrics that a standard smart device can retrieve using its built-in sensors as many as possible. After exploring a device that we would use to collect data in this study (2019 Samsung Galaxy Tab A), we came up with a list of variables that the device was capable to collect as follows:

Figure 6: Bangna - Chon Buri Expressway located above Bangna – Trat Frontage Road (image source: *www.dailynews.co.th*).

1. **Distance** (*distance*)**:** A distance representing how far a vehicle moved within a 2-second interval (we preset the interval). A distance was obtained by computing a difference between two points (start and end) at a certain time tick.
2. **Speed** (*speed*)**:** A speed of a vehicle at a certain point of time. A speed was obtained by a fraction of distance to time (2 second).
3. **Direction** (*direction*)**:** A direction that a car was heading to. It was computed by evaluating an angle α between a line segment $\langle p_s, p_e \rangle$ and a line $y = 0$ where p_s and p_e are a start point and an end point of a certain time interval, respectively.
4. **Light Intensity** (*lux*)**:** A light intensity that can be retrieved by a light sensor which always comes with most smartphone devices.
5. **Time:** A timestamp that consisted of hour (*hour*) and minute (*minute*).
6. **Day-of-Month:** A day of month.
7. **Day-of-Week** (*day*)**:** A day of week.
8. **Bound** (*bound*)**:** A direction bound that informs which side of the road that a vehicle was running on, which can be either eastbound or westbound.
9. **Road Type** (*road_type*)**:** A class variable indicating whether a vehicle is on an elevated road or a ground-level road.

After we had a set of variables ready and stable, we then developed an Android application that retrieved those variables and recorded as a log file (with CSV extension for easily compatible with most statistics and data science software tools such as R or Python). Note that when we finished the data gathering, we decided not to include Day-of-Month in the set of features because we did not have data of every day (of month) equally, which might lead to an imbalance problem that would insignificantly contribute to the model.

3.2 Data Gathering

For the process of data gathering, we selected Bangna - Chon Buri Expressway and Bangna – Trat Frontage Road as an elevated road and a ground-level road, respectively. Both roads lie east-west and link Bangkok, a capital city of Thailand, with Chon Buri, a famous tourist cities in Thailand. The two roads are located on the north of Thai Gulf. Figure 7 shows both roads on Google Maps. To collect data, we drove along the road while enabling the application to automatically get values of all variables from the device's sensors and log them. The application was set to refresh the logging task every two second as previously mentioned.

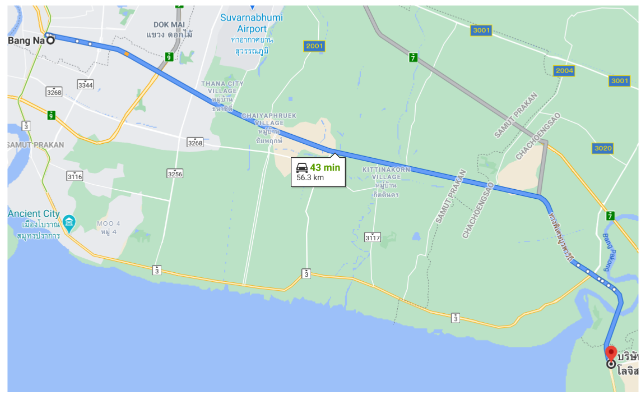


Figure 7: Layout of Bangna - Chon Buri Expressway and Bangna – Trat Frontage Road on Google Maps.

To have data balance in most variables as much as possible, we managed to have a similar number of trips of elevated road & ground-level road, eastbound drive & westbound drive, daytime drive & nighttime drive, and balanced day of week (Sunday to Saturday) since a day of week affects traffic, which might contribute to a prediction model. After data gathering processing, we logged 24362 instances, which can be divided into 11095 instances of the elevated road and 13267 instances of the ground-level road.

Table 1 shows examples of instances of data in the log file comparing with the old one from PatanasakPinyo [11] (Table 2).

3.3 Model Implementation

We inherited a concept from PatanasakPinyo [11] to train a classification model to classify the road type using a Random Forest as a classifier to observe whether there existed any differences when training data were collected from different roads, particularly, Bangna - Chon Buri Expressway and Bangna – Trat Frontage Road, rather than Borommaratchachonnani Elevated Road and Borommaratchachonnani Frontage Road. We decided to use R as a tool to filter and handle pre-processing the raw data. We partitioned the data set into a training set (70%) and a test set

Table 1: Instances of data collected from data gathering process.

year	month	date	day	hour	minute	second	lat	lng	altitude	distance	speed	direction	lux	temp	bound	road_type
2020	5	7	4	12	0	50	13.66709596	100.6385018	-34	3.340153002	6012.275403	356.8788	993	0	E	elevated
2020	5	7	4	12	0	52	13.66698917	100.6389127	-17	0.005308390	9.555102	240.6810	989	0	E	elevated
2020	5	7	4	12	0	54	13.66690162	100.6392806	-3	0.004070347	7.326625	241.4041	979	0	E	elevated
2020	5	7	4	12	0	56	13.66686327	100.6395299	-8	0.005221977	9.399559	251.2549	963	0	E	elevated
2020	5	7	4	12	0	58	13.66670674	100.640143	-10	0.004934684	8.882431	245.6843	981	0	E	elevated
2020	5	7	4	12	1	0	13.66661957	100.6406102	-12	0.005106850	9.192331	246.4602	970	0	E	elevated
2020	5	7	4	12	1	2	13.66652733	100.6411005	-14	0.005265021	9.477038	249.0224	959	0	E	elevated
2020	5	7	4	12	1	4	13.66643043	100.6416091	-13	0.005284162	9.511492	248.0496	948	0	E	elevated
2020	5	7	4	12	1	6	13.66632566	100.6421179	-13	0.005281114	9.506005	244.6641	965	0	E	elevated
2020	5	7	4	12	1	8	13.66621879	100.642636	-12	0.005382215	9.687987	246.2216	944	0	E	elevated
2020	5	7	4	12	1	10	13.66611976	100.6431517	-13	0.005352376	9.634277	248.1123	930	0	E	elevated
2020	5	7	4	12	1	12	13.66602714	100.6436797	-12	0.005392556	9.706601	248.7074	936	0	E	elevated
2020	5	7	4	12	1	14	13.66592526	100.6442047	-13	0.005368668	9.663602	246.5293	941	0	E	elevated
2020	5	7	4	12	1	16	13.66581964	100.6447341	-13	0.005407020	9.732636	246.2482	940	0	E	elevated

Table 2: Instances of data in the preliminary study [11].

year	month	date	day	hour	minute	second	lat	lng	altitude	distance	speed	direction	lux
2020	1	2	4	15	15	14	13.78085367847234	100.43005842715502	-8.0	0.005088622	9.159520	255.885606	366
2020	1	2	4	15	15	16	13.780807997100055	100.4305356927216	-9.0	0.005106809	9.192256	254.043083	362
2020	1	2	4	15	15	18	13.780763824470341	100.43102058582008	-11.0	0.005130625	9.235126	255.199469	235
2020	1	2	4	15	15	20	13.780721286311746	100.4315016232431	-11.0	0.005102782	9.185007	256.565564	327
2020	1	2	4	15	15	22	13.780686459504068	100.43196966871619	-11.0	0.005040076	9.072137	257.220647	399
2020	1	2	4	15	15	24	13.780636419542134	100.4324477724731	-11.0	0.005116951	9.210512	251.996177	448
2020	1	2	4	15	15	26	13.78058788832277	100.43291942216456	-10.0	0.005072030	9.129654	254.004095	406
2020	1	2	4	15	15	28	13.780519072897732	100.43360053561628	-10.0	0.004938547	8.889385	253.396242	423
2020	1	2	4	15	15	30	13.7804801389575	100.4340636357665	-11.0	0.005048775	9.087795	256.795132	422
2020	1	2	4	15	15	32	13.78043420612812	100.43452698737383	-12.0	0.004994445	8.990002	254.453015	555

(30%). The variables that were fed as the model features were day-of-week, hour & minute, distance, speed, direction, light intensity, and direction bound. A class attribute was a type of the road (elevated/ground). The final model was trained in R using Caret Library to handle Random Forest. For train control, we chose to implement a 10-fold cross validation. Since other important training parameters such as $mtry$ (i.e., a number of variables to be randomly sampled as candidates at each split in a tree) were not predefined, we trained the model multiple times to obtain optimal values of those training parameters as well.

4 Results and Discussion

To come up with the most optimal classification model, we needed to assign training parameters to be a certain value that we had not known until we did experiments. Hence, the first half of this section is to report results from experiments to select the best values of each training parameter. We then reported results of training and testing of the classification model along with important statistics.

The first training parameter was $mtry$. An $mtry$ is a number of variables to be randomly sampled as candidates at each split in a tree. We trained a model by using a value from one to eight. After training, we found that $mtry = 3$ is the most optimal.

Next, we trained a model to find the optimal value of *maxnodes*. A *maxnodes* is a maximum number of nodes in a tree. We tried values from five to fifteen. After training, the best value was $maxnodes = 14$, which yielded an accuracy of 0.941349 (Table 3).

Table 3: Accuracy Rates of Different Values of *maxnodes*.

```
Accuracy
       Min.   1st Qu.    Median      Mean
5  0.8533724 0.8617911 0.8660206 0.8661228
6  0.8539589 0.8659824 0.8736440 0.8749770
7  0.8668622 0.8809553 0.8870968 0.8864722
8  0.8826979 0.8912182 0.8947522 0.8943880
9  0.8862170 0.8951613 0.8965115 0.8979657
10 0.8774194 0.8918043 0.8970972 0.8977304
11 0.8938416 0.8996047 0.9047201 0.9036533
12 0.8985932 0.9036657 0.9070660 0.9085798
13 0.9021102 0.9131965 0.9138084 0.9152059
14 0.9043988 0.9083578 0.9167644 0.9176682
15 0.9102639 0.9214076 0.9255357 0.9242368

     3rd Qu.      Max. NA's
5  0.8743402 0.8751465    0
6  0.8854045 0.8961877    0
7  0.8951776 0.8991202    0
8  0.9006015 0.9043988    0
9  0.9049998 0.9079179    0
10 0.9064657 0.9126100    0
11 0.9086646 0.9114370    0
12 0.9108896 0.9284457    0
13 0.9197947 0.9255132    0
14 0.9236183 0.9413490    0
15 0.9291895 0.9366569    0
```

The last one was the number of trees in the forest ($ntree$). We tried values from 100 to 900. After training, the best value was $ntree = 100$, which yielded an accuracy of 0.9431085 (Table 4).

Table 4: Accuracy Rates of Different Values of $ntree$.

```
Accuracy
        Min.   1st Qu.    Median      Mean
100 0.9032258 0.9125020 0.9161534 0.9187237
200 0.9043988 0.9083578 0.9167644 0.9176682
300 0.9008798 0.9127566 0.9152738 0.9171405
400 0.9096774 0.9162879 0.9190858 0.9201316
500 0.9090909 0.9167277 0.9199651 0.9200728
600 0.9108504 0.9180713 0.9205512 0.9214799
700 0.9108504 0.9186694 0.9225806 0.9219492
800 0.9120235 0.9171674 0.9211374 0.9220077
900 0.9108504 0.9177538 0.9214310 0.9227111

      3rd Qu.      Max. NA's
100 0.9231672 0.9431085    0
200 0.9236183 0.9413490    0
300 0.9196952 0.9390029    0
400 0.9217352 0.9384164    0
500 0.9218475 0.9378299    0
600 0.9222874 0.9372434    0
700 0.9225806 0.9378299    0
800 0.9258065 0.9378299    0
900 0.9250733 0.9372434    0
```

After we had optimal values of training parameters ($mtry$, *maxnodes*, $ntree$), we assign those values in the model configuration and trained the final model. The final model yielded an accuracy of 0.9245 with $(0.9182, 0.9304)$ as 95% confident interval when we tested the model with the test set. The test set had 7309 instances of data. The 3086 instances of elevated road were correctly classified while 287 were incorrectly classified as a ground-level road. The 3671 instances ground-level road were correctly classified while 265 were incorrectly classified as an elevated road. The first five important variables sorted by variable importance were lux, *hour*, *speed*, *day*, and *distance*. Figure 8 shows the completed test result (confusion matrix and statistics) generated by Caret.

```
          Reference
Prediction elevated frontage
  elevated     3086      265
  frontage      287     3671

               Accuracy : 0.9245
                 95% CI : (0.9182, 0.9304)
    No Information Rate : 0.5385
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.848

 Mcnemar's Test P-Value : 0.3714

            Sensitivity : 0.9149
            Specificity : 0.9327
         Pos Pred Value : 0.9209
         Neg Pred Value : 0.9275
             Prevalence : 0.4615
         Detection Rate : 0.4222
   Detection Prevalence : 0.4585
      Balanced Accuracy : 0.9238

       'Positive' Class : elevated
```

Figure 8: Confusion Matrix and Statistics after Testing the Model.

We also trained one more model to see how much better a classification model would be if we included an altitude in the feature set regardless of the fact that the accuracy level of an altitude retrieved by a GPS sensor in an average-grade smartphone was not reliable if a difference in height was less than a specific threshold. With this model, we replicated what we did previously. We used the same training and test data sets as well as all presets to ensure that every factor was properly preserved except that the feature set, particularly, we added altitude to it.

With the classification model that included an altitude, we found that $mtry = 2$ was optimal for this case. Next, we looped through various values of *maxnodes* from five to fifteen. It was showed up that *maxnodes* = 15 is optimum. Table 5 shows an accuracy rate for each value of *maxnodes*.

Table 5: Accuracy Rates of Different Values of *maxnodes*.

Accuracy

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
5	0.9407625	0.9585287	0.9683284	0.9640532	0.9712610	0.9759531	0
6	0.9513482	0.9662757	0.9683284	0.9669282	0.9709721	0.9771261	0
7	0.9659824	0.9708211	0.9730363	0.9736120	0.9769795	0.9800587	0
8	0.9700880	0.9725930	0.9756670	0.9758992	0.9788856	0.9818182	0
9	0.9712610	0.9755167	0.9771395	0.9774234	0.9791789	0.9835777	0
10	0.9747801	0.9755240	0.9774326	0.9778340	0.9800587	0.9812317	0
11	0.9712610	0.9750844	0.9765463	0.9775411	0.9809384	0.9841642	0
12	0.9730205	0.9766965	0.9788923	0.9787721	0.9810850	0.9835777	0
13	0.9724340	0.9768361	0.9785986	0.9787136	0.9810850	0.9829912	0
14	0.9736070	0.9771395	0.9788920	0.9795347	0.9818182	0.9853372	0
15	0.9765533	0.9786019	0.9800645	0.9808836	0.9818182	0.9894428	0

Similarly, we lopped through various values of *ntree* and found that the best case was when *ntree* = 100. Table 6 shows an accuracy rate for each value of *ntree* that we tried.

We trained the final classification model that included an altitude one last time using optimal values of training parameters that we just retrieved. After we got the model, we fed the test data set to it to observe a performance. The model yielded an accuracy rate of 0.9793 with (0.9758, 0.9825) as 95% CI. In 3304 instances of elevated road were classified correctly while 69 instances were incorrectly classified. On the other hand, 3854 instances of frontage road were correctly classified while 82 were incorrectly classified. Figure 9 shows test results and related statistics.

The model reported the first five important variables as *altitude*, *lux*, *hour*, *distance*, and *speed*. All of them were overlapped with a set of important variables of the first model (the model without an altitude) except *altitude*. Figures 10, 11, 12, and 13 show box plots of distribution of instances of data (only the training data set) group by each important variable, particularly, *lux*, *hour*, *speed*, and *distance* so the reader can view a behavior of training data. Note that we omitted *day* because it contained factor data.

After comparing both results from the two models (with and without an altitude), we found that having an altitude did not significantly help improving the model's performance.

Table 6: Accuracy Rates of Different Values of *ntree*.

Accuracy

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
100	0.9753810	0.9790445	0.9812317	0.9815875	0.9843109	0.9900293	0
200	0.9765533	0.9786019	0.9800645	0.9808836	0.9818182	0.9894428	0
300	0.9765396	0.9772860	0.9794780	0.9804730	0.9824047	0.9900293	0
400	0.9765396	0.9768464	0.9800643	0.9801798	0.9822581	0.9870968	0
500	0.9759672	0.9783023	0.9797712	0.9802385	0.9824047	0.9859238	0
600	0.9759672	0.9778690	0.9809439	0.9803557	0.9828446	0.9847507	0
700	0.9765396	0.9774326	0.9800643	0.9801797	0.9828446	0.9847507	0
800	0.9753666	0.9778722	0.9800642	0.9801796	0.9826979	0.9853372	0
900	0.9747801	0.9784584	0.9803576	0.9802969	0.9828446	0.9853372	0

```
          Reference
Prediction elevated frontage
  elevated     3304       82
  frontage       69     3854

               Accuracy : 0.9793
                 95% CI : (0.9758, 0.9825)
    No Information Rate : 0.5385
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9584

 Mcnemar's Test P-Value : 0.3288

            Sensitivity : 0.9795
            Specificity : 0.9792
         Pos Pred Value : 0.9758
         Neg Pred Value : 0.9824
             Prevalence : 0.4615
         Detection Rate : 0.4520
   Detection Prevalence : 0.4633
      Balanced Accuracy : 0.9794

       'Positive' Class : elevated
```

Figure 9: Confusion Matrix and Statistics after Testing the Model.

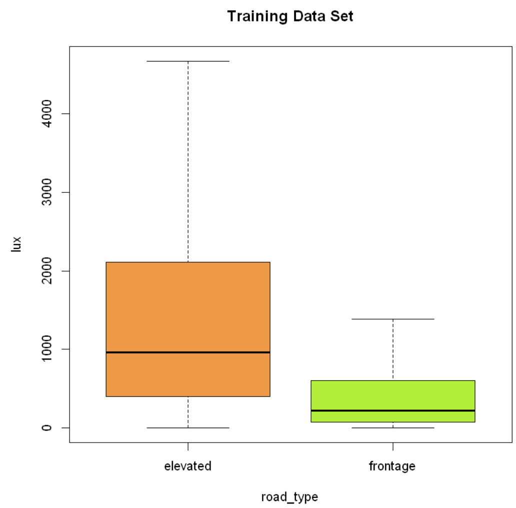


Figure 10: Box plots illustrate a distribution of *lux* of both classes.

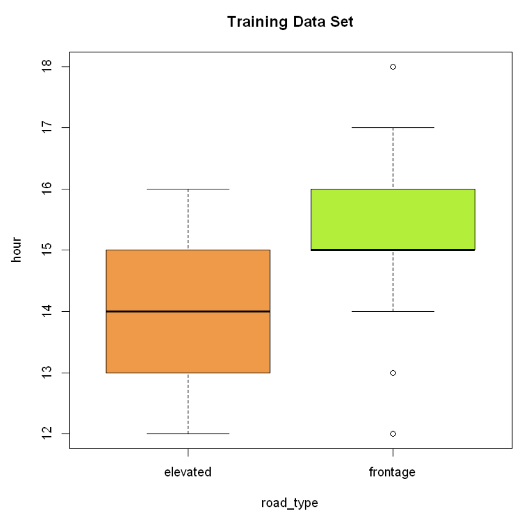


Figure 11: Box plots illustrate a distribution of *hour* of both classes.

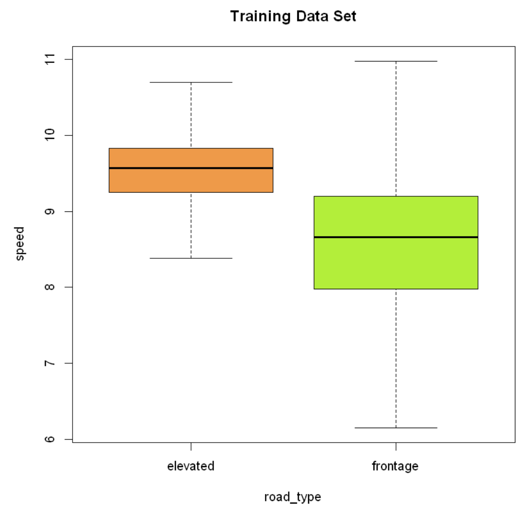


Figure 12: Box plots illustrate a distribution of *speed* of both classes.

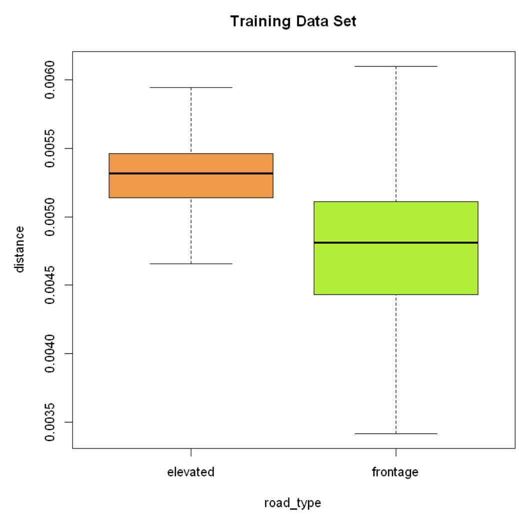


Figure 13: Box plots illustrate a distribution of *distance* of both classes.

5 Conclusions

In this study, we developed a classification model that was capable of classifying the road type that a vehicle was running on if it was an elevated road or a ground-level road where a traditional location-based application cannot do using geolocation data since both roads share exact pairs of a latitude and a longitude all along the way. We inherited a methodology from PatanasakPinyo [11] to train a model using Random Forest as a classifier with 10-fold cross validation as a train control. The training parameters were examined to ensure that they were the most optimal. The data that were used in this study were collected from driving on Bangna - Chon Buri Expressway and Bangna – Trat Frontage Road. Both roads are located in Thailand. The result of testing the model showed an accuracy of 0.9245 with a 95% CI of $(0.9182, 0.9304)$. For future extension of this study, we are going to do a cross-test by testing our classification model on data set used in [11].

Acknowledgments

This research project is supported by both Mahidol University and The Faculty of Information and Communication Technology, Mahidol University.

References

[1] Georgi Batinov, Michelle Rusch, Tianyu Meng, Kofi Whitney, Thitivatr Patanasakpinyo, Les Miller, and Sarah Nusser. "Understanding Map Operations in Location-based Surveys". In *Eighth International Conference on Advances in Computer-Human Interactions (ACHI 2015)*, Lisbon, Portugal. International Academy, Research, and Industry Association (IARIA) pp. 144-149, 2015.

[2] Allen Yilun Lin, Kate Kuehl, Johannes Schöning, and Brent Hecht. "Understanding 'Death by GPS' A Systematic Study of Catastrophic Incidents Associated with Personal Navigation Technologies". In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 1154–1166 2017.

[3] Paul W McBurney and Arthur N Woo. "Infrastructure-aiding for Satellite Navigation Receiver and Method". US Patent 6,473,030 October 29, 2002.

[4] Paul W McBurney and Arthur N Woo. "Satellite Navigation Receiver and Method". US Patent 6,437,734 August 20, 2002.

[5] Joe Mehaffey. "Error Measures". *http://gpsinformation.net/main/errors.htm*.

[6] Joe Mehaffey. "GPS Altitude Readout How Accurate?". *Retrieved*, 12(12):2016, 2001.

[7] Marko Modsching, Ronny Kramer, and Klaus ten Hagen. "Field Trial on GPS Accuracy in a Medium Size City: The Influence of Built-up". In *3rd workshop on positioning, navigation and communication*, 2006:209-218, 2006.

[8] Kent L Norman. "Spatial Visualization—A gateway to computer-based technology". *Journal of Special Education Technology*, 12(3):195–206, 1994.

[9] Eunil Park and Ki Joon Kim. "Driver Acceptance of Car Navigation Systems: Integration of locational accuracy, processing speed, and service and display quality with technology acceptance model". *Personal and ubiquitous computing*, 18(3):503–513, 2014.

[10] Thitivatr PatanasakPinyo. *"Flattening Methods for Adaptive Location-based Software to User Abilities"*. Graduate Theses and Dissertations, Iowa State University, 2017.

[11] Thitivatr Patanasakpinyo. "Ameliorating Accuracy of a Map Navigation When Dealing with Different Altitude Traffics that Share Exact Geolocation". In Alex Redei, Rui Wu, and Frederick Harris, editors, *SEDE 2020. 29th International Conference on Software Engineering and Data Engineering*, EPiC Series in Computing. EasyChair, 76:95–104, 2021.

[12] Thitivatr PatanasakPinyo, Georgi Batinov, Kofi Whitney, and Les Miller. "Methods That Flatten the User Space for Individual Differences in Location-based Surveys on Portable Devices". In *31st International Conference on Computers and Their Applications (CATA 2016)*. International Society for Computers and Their Applications (ISCA), Las Vegas, Nevada, pp. 65-70, 2016.

[13] Thitivatr Patanasakpinyo, Georgi Batinov, Kofi Whitney, Adel Sulaiman, and Les Miller. "Object-Indexing: A Solution to Grant Accessibility to a Traditional Raster Map in Location-Based Application to Accomplish a Location-Based Task". *International Journal of Computing, Communication and Instrumentation Engineering (IJCCIE)*, 5(1):1–5, 2018.

[14] Thitivatr Patanasakpinyo, Georgi Batinov, Kofi Whitney, Adel Sulaiman, and Les Miller. "Enhanced Prediction Models for Predicting Spatial Visualization (VZ) in Address Verification Task". *Proceedings of 34th International Conference on Computers and Their Applications*, 58;247-256, 2019.

[15] Thitivatr PatanasakPinyo, Georgi Batinov, Kofi Whitney, Adel Sulaiman, Les Miller, and Stephen Gilbert. "Extracting Useful Features for Users with Different Levels of Spatial Visualization". In *33rd International Conference on Computers and Their Applications (CATA 2018)*. International Society for Computers and their Applications (ISCA), Las Vegas, Nevada, pp. 86-91, 2018.

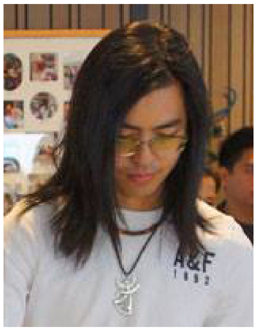
[16] Thitivatr Patanasakpinyo and Les Miller. "UI Error Reduction for High Spatial Visualization Users when Using Adaptive Software to Verify Addresses". In Gordon Lee and Ying Jin, editors, *Proceedings of 35th International Conference on Computers and Their Applications*, EPiC Series in Computing, Easy Chair, 69:22-31, 2020.

[17] Timothy A Salthouse, Renee L Babcock, Debora RD Mitchell, Roni Palmon, and Eric Skovronek. "Sources of Individual Differences in Spatial Visualization Ability". *Intelligence*, 14(2):187–230, 1990.

[18] Pierluigi Silvestrin, Peter Daly, David Walsh, and Eric Aardoom. "Receiver for a Navigation System, in Particular a Satellite Navigation System", May 30 2000. US Patent 6,069,583.

[19] Adel Sulaiman. *"Training and Evaluation in a Large-scale Virtual Environment for a Location-based Mobile Application"*, volume 17573. Graduate Theses and Dissertations, Iowa State University, 2019.

[20] Kofi Whitney. *"Taking the Lab on the Road and Bringing the Road to the Lab: On using mixed-methods and virtual reality to study a location-based task"*, volume 17123. Graduate Theses and Dissertations, Iowa State University, 2019.



Thitivatr PatanasakPinyo is a Lecturer at The Faculty of Information and Communication Technology, Mahidol University, Thailand. He received his PhD with Graduate Teaching Excellence Award from Iowa State University in 2017 under a supervision of Professor Dr. Les Miller and Prof. Dr. Wallapak Tavanapong. His area of interest consists of interface designs for individual difference, location-based systems, and computational theory.